错误处理

错误的分类

分三种：

语法错误：

程序运行之前，都要先检查语法。如语法错误，就会立即报错，并且不会去执行程序。

运行错误：

语法检查通过后开始运行并在此过程中遇到错误。常见的有：

提示性错误：

警告性错误：

致命错误：

逻辑错误：

指的是，程序本身可以正常执行，没有报错——但计算结果却错了。

错误的分级

Php中，将各种错误进行了不同级别的分级归纳，并形成大约有10几个级别的错误，这就是技术层面的错误分级。

每一级别的错误，都是一个“代号”，这个代号其实也就是一个系统内部的“常量”而已。比如：

系统常见错误：

E\_ERROR：致命错误

E\_WARNING:警告性错误

E\_NOTICE:提示性错误

用户自定义的错误：

E\_USER\_ERROR：自定义致命错误

E\_ USER\_WARNING: 自定义警告性错误

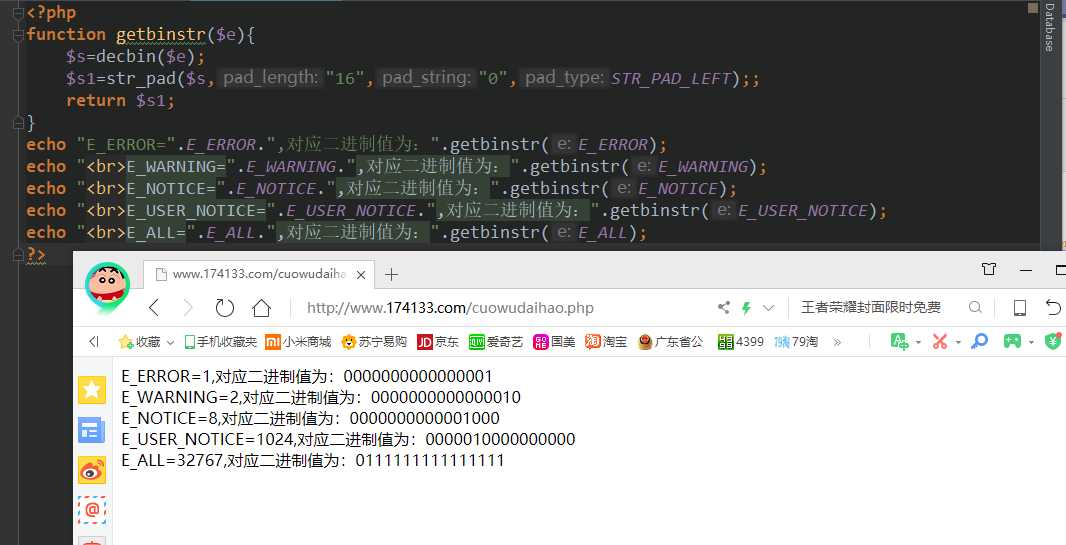
E\_ USER\_NOTICE: 自定义提示性错误

其他:

E\_STRICT 严谨性语法检查错误

E\_ALL 代表“所有错误”。

这些错误代码的实际值



错误触发

就是让错误发生

有两种方式会触发错误：

**系统触发：**

系统运行到某行代码，出现某错误，此时系统就是报错

E\_NOTICE:提示性错误 会提示错误，继续执行后续代码

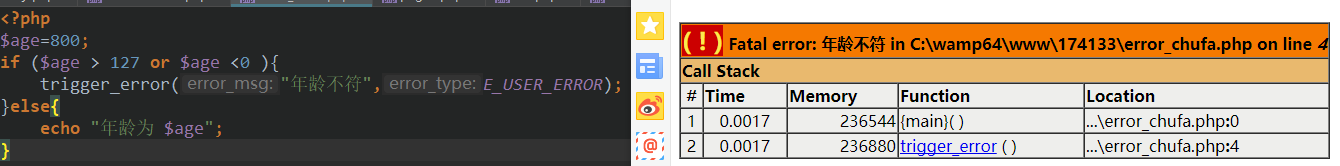
E\_WARNING:警告性错误 会提示错误，继续执行后续代码（也可能看具体情况）

E\_ERROR：致命错误 导致程序无法执行后续语句

比如不存在的函数！

**自定义触发：**

当我们处理某些数据的时候，本来数据本身是没有错误的，但根据具体应用的需要，会要求数据满足某种条件。而该数据并不满足的时候，我们就可以在程序中 主动 去触发（创建）一个错误，以表明该数据的“非法性”；



触发自定义致命错误

Trigger\_error（“错误提示”，三种用户错误代号错误之一）；

其中触发了用户的致命错误（E\_USER\_ERROR）

# 错误报告的显示问题

就是显示在网页上的错误提示内容

有关错误报告，有2个问题需要处理

## 是否提示错误报告（display）：

有两种做法可以来设定是否显示

做法1：

在php.ini 中，设定display\_error的值，显示为no；

注意：前提是条件是我们apache已经装载了php.ini文件

做法2：

直接在php脚本中使用ini\_set（）函数

Ini\_set（display\_errors,0）设置为1就是显示；

注意：

1.不管什么形式，单词都是一样的 display

2.使用php.ini配置是会全局化；

3.在某个脚本代码中使用ini\_set设置，就只影响该代码脚本本身；

4.脚本中设置优先于php.ini配置文件；

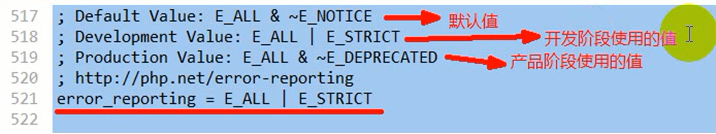
## 显示那些级别的错误报告（error\_reporting）：

前提是“display\_errors”设置为1，表示可以显示。

显示哪些级别的错误报告，也有两个做法：

做法1：

在php.ini中，找到error\_reporting



这个值目前代表所有错误都显示

修改为error\_reporting=E\_NOTTCE；只显示E\_NOTTCE错误

修改为error\_reporting=E\_ERROR | E\_STRICT 真正显示所有错误

做法2：

在脚本代码中：

Ini\_set（“error\_reporting”，E\_NOTICE）显示该级别的错误

Ini\_set（“error\_reporting”，E\_NOTICE | E\_WARNING）显示该2个级别的错误

Ini\_set（“error\_reporting”，E\_ERROR | E\_STRICT）显示所有错误

新版只需要E\_ALL

# 错误日志的记录问题

错误日志其实就是错误报告，只是他会“写入文件中”，此时就称为错误日志！

也有2个问题，每个问题也有2种办法：

**是否记录：（l og\_ error）**

Php.ini中：

Log\_errors=ON 或off

脚本中：

INI\_set（“log\_erros”，1）

补充：  
1.ini\_set（“php配置项”，1）设置值

2.$v1=ini\_get（“php配置项”）获取值

**记录到哪里？？（error\_log）**

一般就只有2个写法：

写法1：直接使用一个文件名，此时系统会自动在每个文件夹下都建立该文件名，并用其记录文件夹下所有网页文件发生的错误信息。

Ini\_set(error\_log“”,”\*.txt”)//记录到指定文件

新建一个\*.txt文件来记录错误信息。

写法2：使用一个特殊的名字“syslog”，则此时所有错误信息都会记录到系统中

Ini\_set(error\_log“”,”syslog”)//记录到系统日志中

跟写法1一样

系统日志文件在控制面板》管理工具》事件查看器》Windows日志》应用程序

# 自定义错误处理器

就是用来处理错误的机器——函数

自定义错误处理，指不需要系统处理，而是我们完全对错误处理

显示和记录 我们来处理

一旦现在就是将错误处理权交给我们（开发者）

做法：很简单的，两步：

1. 设定一个函数

Set\_error\_handler（“f1”）

1. 定义函数

Function f1 （）{

//正常是显示错误报告和记录错误日志

}

图：

